Artificial Intelligence Project 2

0216234 胡瑞中

✓ 背景

這次的第一步是先實作 k-fold strati_ed cross validation tool，將其輸入檔案分類成 k 等分，並將之分 test 檔，data 檔，處理完上述即用老師所給的 C4.5 和我自己寫的 Naïve Bayesian classifier 執行~

✓ 程式架構

(一) CV_tool

==========

首先由於 k 是變數即一開始讓使用者輸入 k 值，而為了在測試方便
char* testdata[] = {"adult","car","isolet","pageblocks","winequality"};
則建立陣列讓它以迴圈執行，並從檔案 names 直接複寫 names，而 data 則分成 k 等分並以除以 k 的餘數為依據做分區，而直接輸出 test，由於我是用 vector 存取，所以當一行輸出至 test 時我就將之從 vector 中 erase，最後將剩下的直接輸出 data。

(二) NB

==========

如上一開始先從變數 k 先輸入拿值且所有檔皆由 strtok 函數做拆解

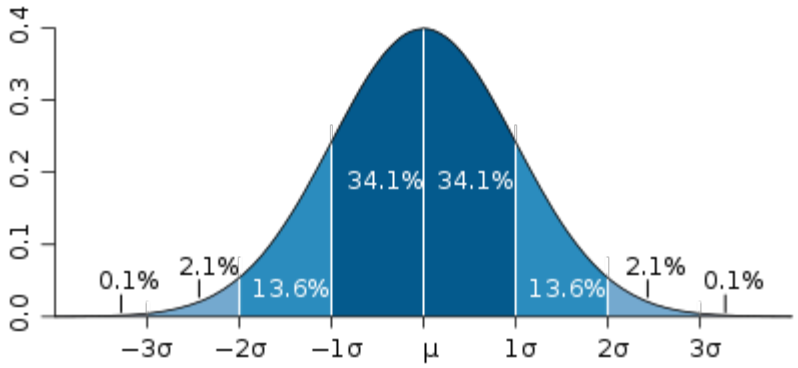
1. 分析 names 檔

   由於觀察到第一行即為最後 class 的屬性則將它 push 到名為 cate 的 vector 中，而剩下即 push 到 text 中，在觀察字串裡是否包含 continuous，若有就”標記”是哪個 attribute。

2. 存取 data 檔及解析

   宣告二維 vector 陣列 data，並將 fgets 到的字串做 strtok 函數拆解並存入 data 中，但由於並不是所有 attribute 都像 car 檔一樣只有離散數據，所以先從.name 檔中得知的”標記”即我所定義的 continuous 陣列。

   再來是解決連續性數據定義為離散區間的方法，我從以前高中學的常態分佈的信賴區間開始著手，求出所有的平均數、標準差和變異數，已從平均數為基準設加減一個和兩個標準差，則區分出總共 6 個區域 A、B、C、D、E、F，將原本連續性數據變為英文字。



3. 測試 test 檔學習並 output 成功率

在存取 test 檔時，並處理連續性數據改為離散區間。

再來就是依照樸素貝葉斯分類器的定義，先求出所有可能 class 會發生的機率而就是在 test 檔所給的 attribute 的情況下會有這 class 發生的機率，並比較哪個可能性機率最大，最後 output 出自己預測的跟正確解答的答對率。

✓ 結果探討

■ Result_template

dataset		cv1	cv2	cv3	cv4	cv5	cv6	cv7	cv8	cv9	cv10	avg	p-value
adult	NB_Acc	0.78	0.78	0.78	0.80	0.80	0.77	0.78	0.79	0.79	0.78	0.79	3.531E-09
	c4.5_Acc	0.85	0.86	0.86	0.87	0.86	0.86	0.86	0.85	0.85	0.86	0.86	
car	NB_Acc	0.61	0.63	0.61	0.66	0.63	0.59	0.60	0.61	0.58	0.60	0.61	5.393E-12
	c4.5_Acc	0.93	0.95	0.94	0.94	0.93	0.95	0.92	0.96	0.92	0.94	0.94	
isolet	NB_Acc	0.86	0.90	0.91	0.88	0.92	0.88	0.88	0.90	0.90	0.85	0.89	4.602E-06
	c4.5_Acc	0.80	0.79	0.78	0.83	0.81	0.77	0.79	0.78	0.76	0.80	0.79	
page-blocks	NB_Acc	0.88	0.84	0.85	0.87	0.86	0.84	0.87	0.84	0.86	0.83	0.85	4.289E-10
	c4.5_Acc	0.97	0.97	0.96	0.97	0.99	0.97	0.97	0.97	0.98	0.96	0.97	
winequality	NB_Acc	0.19	0.18	0.13	0.18	0.15	0.21	0.15	0.17	0.19	0.19	0.17	7.927E-12
	c4.5_Acc	0.59	0.64	0.58	0.59	0.62	0.59	0.58	0.60	0.62	0.57	0.60	

如果先不論 winquality 檔，在我寫的 NB 和老師所附的 C4.5 所的出的結果相差不遠，當然我所寫的 NB 必定有蠻多不周全的地方，所以正確率大部分較 C4.5 所來的低，但在 winquality 檔中卻相差很大，而 winquality 檔和 isolet 檔中一樣的都是連續性數據，而在 NB 學習預測是依靠貝氏條件機率的原理，而在貝氏定理下最要求特性之間的獨立性，則在 isolet 檔中是互相獨立的而正確率就明顯的比相依性高的 winquality 檔高出許多。

✓ 過程中的難題

■ Runtime error

原先我的 NB 是像跟 CV_tool 一樣把所有資料跑完，但發現我 NB 演算法寫得不太甚好，一次跑完所有測資會出現 Runtime error，所以我才改成以使用者輸入測資名而跑單一項數據

■ Laplace Estimator

而在寫 NB 時在計算發生某 class 發生機率時一直發生錯誤，當時百思不知其解，將程式一步步 debug 後，才發現 Laplace Estimator 的重要性，
當條件機率連乘時發生乘 0 的情況，使得某類別的計算結果為 0，當然會發生這種如果某個屬性的資料，並不是在每個類別都會出現的情況，也許是我 CV_tool 分切不是很好所導致。

■ 寫 C/C++所遇到的小問題

✧ Vector 二維使用

由於要儲存不知數據會有多大的檔案，而我又想將之放置在類似二維陣列的空間中，就一直思索比較便利輕鬆的方法，
最後我試著用 vector<string> 當一層 row 讓他能用 pushback 函數去存 data 裡一行字串的分割後而成的小字串，再用 vector<vector<string>>去包住原先一層 row 的 vector，而建成我所想要的二維 vector。

✧ 不同型態的問題

在一開始 char* 和 string 的互換算是第一道關卡，
而又當大致寫完所有演算法時，卻卡在最後計算機率的地方，真的超嘔@@，在 int 及 double 中計算發生許多問題，最後則索性將所要計算的變數都設為 double 才得以解決。